

Realitatea Virtuală (Virtual Reality), tehnologie modernă a informaticii aplicate

Marin VLADA – Universitatea din București
Mircea POPOVICI – Universitatea “Ovidius” Constanța
(vlada@fmi.unibuc.ro , dmpopovici@univ-ovidius.ro)

Abstract

*Realitatea Virtuală (Virtual Reality) se referă la un sistem de concepte, metode și tehnici care se utilizează la elaborarea și construirea de produse software în scopul utilizării lor prin intermediul unor sisteme de calcul moderne (calculatoare și echipamente specializate). Acestea oferă modalitatea prin care calculatorul și echipamentele specializate modifică modul în care omul percepe realitatea din mediul natural, prin simularea/modelarea unei alte realități. Se prezintă modelarea în mediile virtuale și produsele software **OpenGL** și **VRML** de creare și vizualizare a scenelor .*

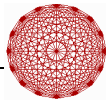
1. Introducere în Virtual Reality

Cuvântul « *Virtual* » este folosit frecvent în studiile / cercetările / experimentele / produsele software ale informaticii și ale domeniului IT&C, în general pentru a desemna o entitate/un mediu care simulează/modelează o entitate/un mediu din realitatea obișnuită a omului. Prin « *realitate* » vom înțelege mediul natural perceput de om prin intermediul simțurilor. Faptul că « *realitatea* » din mediul natural este percepută de om prin intermediul simțurilor, face posibilă simularea / modelarea acesteia prin generarea



/furnizarea datelor/informațiilor percepute de unul sau mai multe simțuri. Din aceste motive, *Realitatea Virtuală (VR-Virtual Reality)* se referă la un sistem de concepte, metode și tehnici care se utilizează la elaborarea și construirea de produse software în scopul utilizării lor prin intermediul unor sisteme de calcul moderne (calculatoare și echipamente specializate). Acestea oferă modalitatea prin care calculatorul și echipamentele specializate modifică modul în care omul percepe realitatea din mediul natural, prin simularea/modelarea unei alte realități. Se poate afirma că acest sistem/mediu, această realitate simulată pe calculator este *Realitatea Virtuală*.

Termeni ca *Virtual Memory*, *Virtual Machine*, *Virtual Server*, *Virtual Private Network (VPN)*, *Virtual University Education (VUE)* se utilizează frecvent în informatică și în domeniul IT&C. De exemplu, termenul *Virtual Memory* se utilizează la simularea memoriei principale a unui sistem de calcul printr-o memorie pe *hard disk*,



astfel că utilizatorul poate să folosească o memorie de capacitate mult mai mare decât cea reală.

În ultimii ani, dezvoltarea explozivă a *tehnologiei microprocesoarelor* a produs pe piața de calculatoare *mașini* din ce în ce mai puternice. Aceste *mașini* sunt echipate cu plăci grafice mai rapide și mai performante, la un preț din ce în ce mai scăzut, devenind astfel posibil, chiar și pentru un utilizator ocazional, să utilizeze produse software ale *graficii pe calculator* (**Computer Graphics**). Fascinația unei *noi realități* începe, deseori, cu jocurile pe calculator și crește la nesfârșit. Ea permite să privim lumea înconjurătoare într-o altă dimensiune și să experimentăm lucruri care, altfel, nu ne-ar fi accesibile în viața reală, sau care nici măcar nu s-au creat încă. Mai mult, lumea *graficii 3D* nu are nici frontiere și nici constrângeri și poate fi creată și manipulată de orice utilizator după dorință. Nu este lipsit de interes să ne gândim pentru o clipă asupra faptului că această lume, este înzestrată cu o a patra dimensiune, aceea a imaginației noastre. Dar, indiferent de stadiul de dezvoltare atins, acesta nu este suficient, utilizatorii dorind tot mai mult. Ei doresc să pășească în această lume și să interacționeze cu ea, în loc să privească doar o imagine pe monitor. Această tehnologie, din ce în ce mai populară și modernă, se numește **Realitate Virtuală** (*Virtual Reality*).

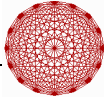
Noile echipamente **VR** încearcă să reproducă funcțional comportamentul normal al omului într-o altfel de realitate: *realitatea virtuală*. Aceste echipamente sunt:

- *ochelari 3D, căști VR (HMD-Head Mounted Display), monitoare 3D;*
- *mănuși VR, volane, gamepad-uri (toate cu “force feedback”, adică sunt capabile să comunice bidirecțional cu mediul virtual);*
- *trackere (acestea urmăresc mișcările corpului uman).*

Aplicațiile cele mai importante ale *realității virtuale* sunt considerate următoarele:

- **modelare, simulare și vizualizare în domeniul științific**, prin care se obțin imaginea și studiul diferitelor modele sau fenomene inaccesibile observației directe (*fluxuri de informații, structuri atomice, sisteme meteorogice, sisteme cosmice etc.*); utilizarea rezultatelor în *software educațional*;
- **experimente și simulări în domeniul medicinei**, pentru învățarea diferitelor proceduri fără riscul vieții pacientului (de exemplu, în *chirurgie*);
- **sisteme de simulare (simulatoare)** pentru antrenamentul *piloților, astronauților, șoferilor etc.*, prin care se pot exersa manevre dificile, fără a fi în pericol viața participantului sau securitatea cabinei de vehicul (avion, elicopter, mașină, tren, navă maritimă, navă spațială etc.);
- **proiectare asistată de calculator (CAD)** în diferite domenii (*construcții, arhitectură etc.*), prin care proiectantul are posibilitatea să vadă rezultatele *proiectului* sub forma imaginii acestuia în timp real, să observe detaliile, să studieze respectarea diverselor criterii, să ia decizii de modificare a parametrilor înainte de construirea prototipului;
- **realizarea de jocuri distractive pe calculator și filme de animație.**

Calculatorul este acela care a permis crearea unor “*medii virtuale*” capabile să reproducă din ce în ce mai fidel *realitatea înconjurătoare*, atât prin intermediul programelor elaborate de specialiști informaticieni (*software specializat* pentru simulatoare, jocuri, programe de interacțiune: *3DWebmaster; WorldUp; CALVIN; 3D Impact ! Pro; 3DAnywhere; Adobe AtmosphereTM; Anark Studio; AXEL; Effect3D; Kazoo 3D*), cât și prin intermediul *echipamentelor și tehnologiilor VR* care i se pot



asocia. *Mediile virtuale* necesită mai multe resurse decât o face un *sistem desktop* standard. Pentru asigurarea interacțiunii cu utilizatorul sunt necesare **dispozitive hardware** de *input/output* suplimentare, împreună cu *driver-urile* speciale asociate. Figura următoare ilustrează cele mai importante părți ale interacțiunii *om-calculator*. Utilizatorul este echipat cu un **HMD** (*Head Mounted Display*), *tracker* și optional, cu un dispozitiv de manipulare (*mouse 3D*, *mănușă VR* etc). Pe măsură ce utilizatorul efectuează acțiuni de deplasare, rotire a capului, deci schimbări ale punctului de vedere, informații care descriu comportamentul acestuia sunt trimise calculatorului prin intermediul *dispozitivelor de intrare*. Calculatorul procesează informația în timp real și generează un *feedback* corespunzător, care este trimis utilizatorului prin intermediul *ieșirilor* (*video*, *audio*, *haptice* etc).

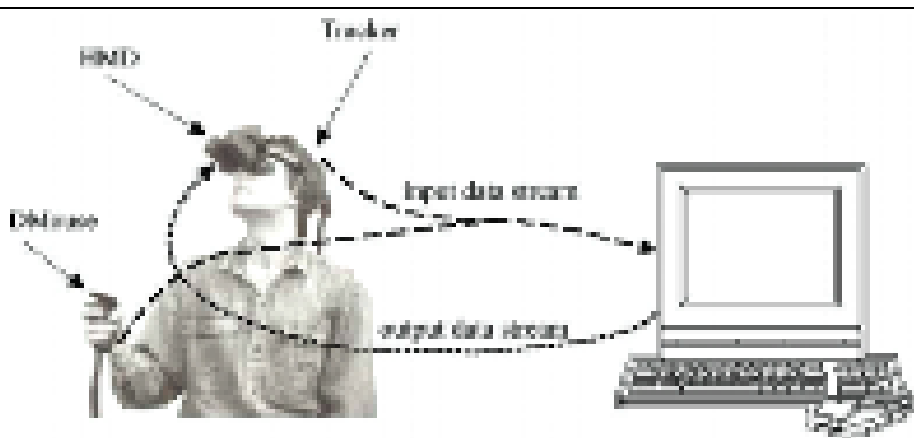
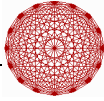


Figura 1. Comunicarea utilizator-mediu virtual

Realitatea virtuală este o simulare generată de calculator a unui *mediu tridimensional*, în care utilizatorul este capabil să vizualizeze și să manipuleze conținutul acestui mediu. Dacă termenul de *multimedia* se referă la date preasamblate și preprogramate, incluzând o suită de informații din anumite medii, *realitatea virtuală* este *dinamică* și în *interacțiune permanentă* cu receptorul/utilizatorul ei. *Multimedia* este la bază bidimensională, o serie de imagini prezentându-se, conform unui scenariu predefinit, pe ecran, pe când **realitatea virtuală** este **tridimensională**, mult mai flexibilă și intens interactivă, combinație avansată de *hardware* și *software multimedia*. Utilizatorul unui *sistem virtual* are libertatea de a explora lumea creată de calculator și de a interacționa direct cu ea.

Conceperea unui *univers tridimensional virtual* se poate realiza folosind **VRML** (**Virtual Reality Modeling Language**), scopul inițial al limbajului fiind tranziția de la o interfață text tip **Web** la una având trei dimensiuni, în permanentă interacțiune cu utilizatorul. **VRML** a fost creat în primăvara anului 1994 și prezentat la prima Conferință **WWW** din Geneva, la dezvoltarea sa avându-se în vedere independența de platformă, extensibilitatea și abilitatea de a lucra în cadrul rețelelor. După specificația inițială VRML 1.0, în anul 1995 a apărut VRML versiunea 1.1, iar în anul 1997 a apărut versiunea VRML 2.0 fiind standard **ISO** și care se numește **VRML97**. Astăzi, **VRML** este rescris în termeni **XML** (**Extensible Markup Language**), noua versiune



având numele **X3D**. În loc de a parcurge pagini cu imagini statice și de a urma *hiperlegături*, utilizatorii pot, de exemplu, să parcurgă coridoare și să manipuleze obiecte, folosind o *casă specială de vizualizare (Head-Mounted Display-HMD)* și o *mănușă VR* pentru „comunicare” cu mediul. Mai nou, a apărut *ecranul retinal virtual (Virtual Retinal Display-VRD)* pentru o explorare mai facilă a lumii **3D**. Informații despre VRML se pot găsi la adresa <http://www.vrml.org>. De asemenea, pentru informații privind *termeni* din domeniul *informaticii* se poate consulta *The Free On-line Dictionary of Computing* la adresa <http://wombat.doc.ic.ac.uk>.

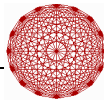
Realitatea Virtuală (RV) și *Mediu Virtual (MV)* sunt termeni utilizați în comunitatea Computer Science. De asemenea, la fel de importanți sunt: *Lume Virtuală (Virtual World)*, *Mediu Virtual (Virtual Environment)*, *Mediu Sintetic (Synthetic Environment)*, *Lume Artificială (Artificial World)*, *Realitate Artificială (Artificial Reality)*. Toate acestea înseamnă, de fapt, concepte care generează următoarele *definiții* pentru **Realitatea Virtuală**:

- “Realitatea Virtuală este un sistem folosit pentru a crea o lume artificială pentru un utilizator astfel încât acesta să aibă impresia că se află în această realitate în care se poate mișca și interacționa cu obiectele înconjurătoare” [C. Manetta, R. Blade-1995]
- “Grafica interactivă în timp real cu modele 3D combinată cu o tehnologie de afișare care oferă utilizatorului imersiunea în modelul lumii și posibilitatea manipulării directe a acestuia” [H. Fuchs, G. Bishop-1992]
- “Iluzia participării într-un mediu sintetic în locul observării externe a acestui mediu. **RV** se bazează pe display-uri 3D stereoscopice purtate de utilizator, urmărite, urmărirea mâinilor/corpului și un sunet binaural. **RV** este o experiență imersivă, multi senzorială” [M. Gigante-1993]
- “Simulări pe calculator care utilizează o grafică 3D și astfel de dispozitive, cum sunt DataGlove, pentru a permite utilizatorului să interacționeze cu simularea” [Jargon Dictionary-1995]
- “Realitatea Virtuală se referă la medii imersive, interactive, multi-senzoriale, centrate spre utilizator, tridimensionale, generate de calculator și combinarea tehnologiilor necesare construirii acestor medii” [C. Cruz-Neira-1993]
- “Realitatea Virtuală ne permite să navigăm și să vedem o lume în trei dimensiuni în timp real, cu șase grade de libertate, fiind, în esență, o clonă (virtuală) a realității fizice” [L. and E. Schweber-1995].

Sistemele de Realitate Virtuală se disting în mai multe categorii:

- sisteme de realitate virtuală imersive (*immersive VR*);
- sisteme de simulare (*simulation VR*);
- sisteme proiective (*projected VR*);
- sisteme cu teleprezență (*telepresence VR*);
- sisteme de realitate îmbogățită (*augmented reality VR*);
- sisteme de realitate virtuală desktop (*desktop VR*).

Imersivitatea este proprietatea sistemului de realitate virtuală care provoacă senzația de „prezență” în *mediul virtual*, adică utilizatorul/participantul se simte în „locul” din mediul virtual afișat pe un display, acesta făcând parte integrantă din mediul virtual care simulează o realitate. Din aceste motive, un sistem de realitate virtuală este compus din mai multe *subsisteme* ce comunică între ele pentru redarea *interacțiunii*.

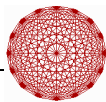


“There is an ever expanding set of interactive graphics applications which require smooth and fast free space tracking of some part of the user's body, or some hand-held object. Headmounted displays (HMDs) for immersive virtual environment simulations have stimulated a tremendous amount of activity since the early 1990s. Many virtual prototyping systems were developed, often using "goggles and gloves" for interaction. While the media has been distracted by the new phenomenon of the world-wide web, virtual environment technology has made great strides, especially in the area of real-time rendering on affordable hardware, and has been silently catapulted out of the laboratory and into real-world applications. Recently, there has been considerable interest in wearable Augmented Reality (AR) systems and virtual set generation for television studios. While these seem to present fairly dissimilar tracking problems (tracking a headset v. tracking a camera), they both require a long-range tracking solution with very high accuracy that will work reliably in an uncontrolled environment full of interference sources” (Eric Foxlin, Michael Harrington, George Pfeifer, *Constellation: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications*, Proceedings of SIGGRAPH 98 -Orlando, Florida, July 19-24, 1998, Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH).

2. Modelare și interacțiune în mediile virtuale

Care sunt modelele și mijloacele de reprezentare ale lumii reale? Soluțiile tehnologice ale **RV** vizează toate simțurile umane, omul fiind elementul central în dialogul om-mașină. În acest sens, există soluții de interfață vizuală, audio, de urmărire, de intrare primară a utilizatorului (cu ajutorul mânușilor VR , exoskeleton, joystick, trackball, 3Dmouse , pen-based), haptică, de mișcare a întregului corp și chiar olfactivă. Rolul *interfețelor vizuale* este acela de cuplare vizuală între utilizator și mediul virtual, sau chiar de cuplare între utilizator și mediul virtual pe de o parte și cel real pe de altă parte, în cazul mediilor reale îmbunătățite, *Augmented Reality(AR)*. În timp ce urmărirea utilizatorului este un tip de interfață care este aproape transparentă utilizatorului, informațiile oferite de acesta (legate de poziția și orientarea utilizatorului) mediului virtual sunt critice în informarea *sistemului* de **RV** asupra poziției și orientării utilizatorului sau, astfel încât să se realizeze o corelare între imaginea recepționată de utilizator și poziția și orientarea acestuia. Interfețele audio pot juca un rol important în oferirea intrărilor informaționale către utilizator, ducând la creșterea *realismului* mediului simulat, și sugerând utilizatorului senzația prezenței în cadrul acestui mediu.

“Modeling and rendering of natural scenes with thousands of plants poses a number of problems. The terrain must be modeled and plants must be distributed throughout it in a realistic manner, reflecting the interactions of plants with each other and with their environment. Geometric models of individual plants, consistent with their positions within the ecosystem, must be synthesized to populate the scene. The scene, which may consist of billions of primitives, must be rendered efficiently while incorporating the subtleties of lighting in a natural environment. Synthesis of realistic images of terrains covered with vegetation is a challenging and important problem in computer graphics. The challenge stems from the visual complexity and diversity of the modeled scenes. They include natural ecosystems such as forests or grasslands, human-made environments, for instance parks and gardens, and intermediate environments, such as lands recolonized by vegetation after forest fires or logging. Models of these ecosystems have a wide range of existing and potential applications, including



computer-assisted landscape and garden design, prediction and visualization of the effects of logging on the landscape, visualization of models of ecosystems for research and educational purposes, and synthesis of scenes for computer animations, drive and flight simulators, games, and computer art" (Oliver Deussen, Patrick Hanrahan, Matt Pharr, Bernd Lintermann, Radomir Mech, Przemyslaw Prusinkiewicz, *Realistic Modeling and Rendering of Plant Ecosystems*, SIGGRAPH 98, Orlando, Florida, July 19-24, 1998, COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998).

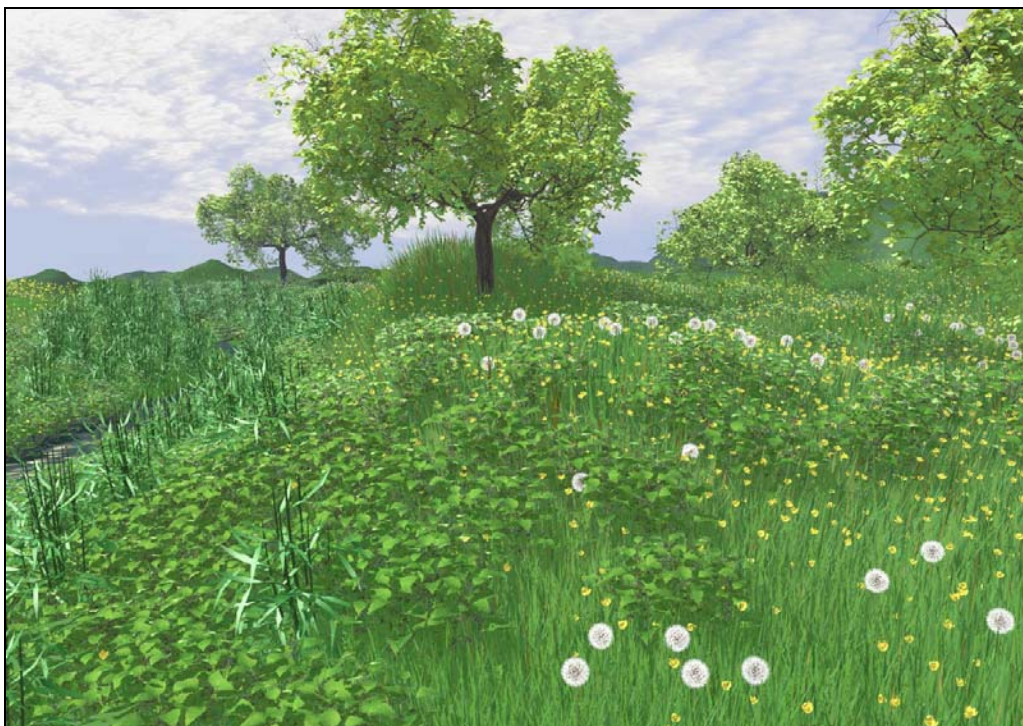
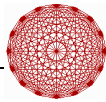


Figura 2. Modelarea virtuală a unui peisaj natural (SIGGRAPH 98)

Generarea de **lumi virtuale** implică o serie de aspecte care nu au fost prezente în *grafica pe calculator* standard. Cea mai mare provocare este performanță versus *comportamentul și aspectul natural* al mediului. Aceste cerințe sunt contradictorii: *modelele* mai convingătoare și *simulările fizice* mai bune necesită mai multe *resurse hardware* și *software*, și deci un cost computațional mărit care va afecta performanța globală. Poate fi imaginată o serie întreagă de **tipuri de modele** reprezentând **lumi virtuale**, începând de la *modelele simple*, cum ar fi o simplă *cameră*, nemobilită, până la *modelele complexe*, cum ar fi un întreg oraș, cu clădirile sale conținând camere, fiecare modelată cu un mare grad al detaliului. În timp ce este trivial să se afișeze un model simplu, cu o performanță adecvată, nu la fel de simplă se prezintă situația în care trebuie să se afișeze milioane de poligoane, chiar dacă sunt încărcate în memorie. Astfel, nu va fi niciodată tehnic posibil, să echilibrăm această balanță, căci, cu cât va fi mai rapid *hardware-ul*, cu atât mai *complexe* și mai rafinate vor fi *modelele*. Este nevoie să se dezvolte **structuri de date** și **algoritmi dedicați** care să permită producerea unei imagini de o calitate foarte ridicată la un cost acceptabil.



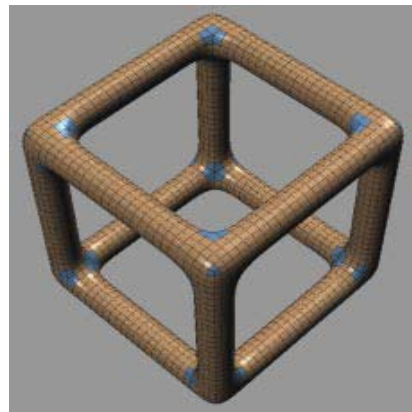
Orice aplicație de **RV** trebuie să prezinte o performanță și interacțiune reale. Această cerință, fundamentală poate fi îndeplinită doar atunci când toate componentele sistemului de intrare, de interacțiune și de ieșire, sunt integrate în mod corespunzător. Din păcate, nici chiar cel mai puternic hardware nu poate atinge singur acest deziderat, el necesitând asistență *software* pentru precizia controlului, managementul resurselor și sincronizare.



Construcția și întreținerea **mediilor virtuale** pe baze fizice, *multi-user* și deci *distribuite*, nu este o sarcină ușoară. Este necesară *independența hardware*, *flexibilitate* și *paradigme* de nivel înalt în vederea programării ușoare, întreținerii și o interfață consistentă cu utilizatorul, pentru a se asigura minimizarea întârzierilor. Câteva din exemplele de **Toolkit-uri** și **sisteme de medii virtuale** sunt: **MR**(Minimal Reality) [Shaw93a,b], **NPSNET** [Zyda92a,b,Mace95], **DIVE** [Carl93], **CAVE** [Cruz93b], **AVIARY** [Snow94], **RING** [Funk95a], **GNU/MAVERIK** [Hubb01].

“Subdivision surfaces have emerged recently as a powerful and useful technique in modeling free-form surfaces. However, although in theory subdivision surfaces admit local parametrizations, there is a strong belief within the computer graphics community that these parametrizations cannot be evaluated exactly for arbitrary parameter values”

(**Jos Stam**, *Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values*, **SIGGRAPH 98**, Orlando, Florida, July 19-24, **1998**, **COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998**).



Elementele principale în conceperea și elaborarea unui *mediu virtual* sunt:

- **componenta grafică 3D** cu ajutorul căreia se realizează vizualizarea;
- **modelul mediului virtual** împreună cu toate elementele care-l populează;
- **motor de control al deplasării** entităților componente;
- **feedback în timp real** din partea ambelor părți participante în cadrul interacțiunii om-calculator;
- **motor de simulare/control** al comportamentului entităților sintetice și fenomenelor naturale desfășurate în mediul curent.

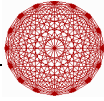
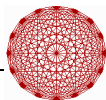


Figura 3. A conceptual sketch of the office of the future

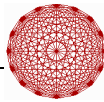
(Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, Henry Fuchs, *The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays*, SIGGRAPH 98, Orlando, Florida, July 19-24, 1998, COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998).

Componentele și funcțiile principale ale unui sistem de RV pot fi împărțite în:

- **proces de intrare** - controlează dispozitivele utilizate în introducerea datelor. Există o mare varietate de astfel de dispozitive: tastatură, mouse, trackball, joystick, trackeri de poziție 3D & 6D (mănușă, head tracker, costum, etc.). Un sistem de RV în rețea va adăuga intrările recepționate din rețea. Un dispozitiv de recunoaștere vocală constituie o îmbunătățire a unui mediu de RV, în special atunci când mâinile utilizatorului sunt ocupate cu alte sarcini;
- **proces de simulare** - proces care cunoaște entitățile componente mediului și intrările care acționează asupra lor; tratează interacțiunile, acțiunile scriptice ale obiectelor, simulările legilor fizice, reale sau imaginare, și determină starea mediului. Această simulare este, de fapt, un proces discret care este iterat la fiecare interval de timp;



- **proces de randare** - *procesele de randare* ale unei aplicații de **RV** sunt cele care creează senzațiile cele mai intense utilizatorului. În cazul unei soluții orientate spre rețea, ieșirile aplicației pot fi destinate altor procese situate în cadrul rețelei. În acest caz, ar trebui să existe procese separate de randare pentru fiecare din *sistemele vizuale, audio, haptice* (touch/force), perceptuale. Fiecare din aceste randari trebuie să posede o descriere a stării lumii provenită din procesul de simulare sau derivând direct din baza de date a mediului, pe parcursul fiecărui interval de timp;
- **baza de date a lumii virtuale** - *lumea virtuală* trebuie definită în cadrul unui *spațiu al lumii*; prin natura sa, ca *simulare pe calculator*, această lume este, în mod necesar, limitată. Calculatorul trebuie să plaseze o *valoare numerică* în locațiile fiecărui punct al fiecărui *obiect* din cadrul lumii. Frecvent, aceste coordonate sunt exprimate în *coordonatele carteziane* x, y, z (*lungime, înălțime, adâncime/cota*), fără însă a exclude posibilitatea utilizării sistemelor de coordonate alternative, cum este cazul celor *sferice* sau *cilindrice*, cele carteziane fiind însă întâlnite în absolut toate aplicațiile. Conversiile dintre sistemele de coordonate sunt, din punct de vedere matematic, relativ simple, dar consuma timp. Baza de date a lumii mai poate conține și informații asupra *controalelor hardware* și modul în care acestea sunt integrate în aplicație.
- **proces de stocare și modelare** - elementele primordiale care sunt stocate în *baza de date a lumii* sunt obiectele care populează lumea, *scripturile* care descriu *acțiunile* acestor obiecte sau ale utilizatorului, deci lucruri care i se întâmplă acestuia, *sursele de lumină, controalele aplicației* și *starea* dispozitivelor hardware; există o serie de metode de stocare a informației despre o lume virtuală: într-un singur *fișier*, o *colecție de fișiere* sau chiar o *bază de date*. Soluția oferită de fișiere multiple este una din cele mai frecvent utilizate de pachetele de dezvoltare a *aplicațiilor de RV*. Astfel, fiecare obiect are unul sau mai multe fișiere (*geometrie, scripturi, etc.*) și există un fișier care reține informația despre întreaga lume care declanșează încărcarea tuturor celorlalte fișiere. Unele sisteme includ și un *fișier de configurare* care definește *conexiunile* cu *interfața hardware*. Alteori, întreaga bază de date este încărcată în timpul pornirii aplicației, în timp ce alte sisteme citesc doar fișiere strict necesare, încărcarea celorlalte realizându-se *dinamic*. O bază de date *orientată spre obiecte* ar fi foarte potrivită unei aplicații de *RV*, chiar dacă această soluție nu este, încă, larg răspândită. Fișierele de date sunt cel mai adesea stocate în *format ASCII*. Obiectele dintr-un mediu virtual pot avea *geometrie*, se pot *organiza ierarhic*, se pot manifesta, eventual prin *scripturi*, și pot avea o serie de alte *atribute*. Esența obiectelor poate avea un puternic impact asupra structurii și proiectării sistemului. *Modelarea formei și geometriei* obiectelor este un domeniu deosebit de vast și divers. Unele abordări caută să obțină o *modelare geometrică* foarte exactă a obiectelor din lumea reală, în timp ce altele creează reprezentări simplificate. Majoritatea *sistemelor de RV* sacrifică detaliu și exactitate în favoarea simplității, pentru a obține o viteză sporită în randare. *Modelarea solidelor* și operațiile booleene, eventual regulate, de modelare, constituie forme de modelare care utilizează *obiecte primitive* (cuburi, sfere, conuri etc.).

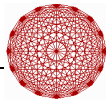


Tony DeRose, Michael Kass, Tien Truong,
Subdivision Surfaces in Character Animation,
SIGGRAPH 98, COMPUTER
GRAPHICS Proceedings, Annual
Conference Series, 1998

**Gabriel Taubin, André Guezic, William
Horn, Francis Lazarus,** *Progressive Forest
Split Compression,* **SIGGRAPH 98,**
COMPUTER GRAPHICS Proceedings,
Annual Conference Series, 1998

Figura 4. Exemple de modelare, reprezentare și vizualizare

Pentru programarea aplicațiilor grafice complexe necesare în crearea și redarea *scenelor virtuale* din mediile virtuale, se utilizează *biblioteci* și *interfețe grafice*, *sisteme de dezvoltare de programe (Toolkit-uri)*, care oferă programatorului /proiectantului posibilitatea să reutilizeze un număr mare de funcții grafice deja elaborate și implementate și, astfel să se concentreze mai eficient asupra proiectării și complexității aplicației. *Limbajele* destinate **RV** oferă diverse facilități și interfețe în vederea proiectării și elaborării de aplicații. Unele oferă o vedere de înalt nivel, în cadrul căreia utilizatorul își poate crea întreaga aplicație, utilizând *limbaje, scriptice* sau nu, și chiar *ustensile grafice*, sistemul fiind responsabil cu întreaga *simulare, geometrie și interacțiune*. Altele se bazează pe o anumită configurație grafică, utilizând **API-uri (Application Programming Interface)** și chiar alte *limbaje de programare*, pentru a asigura o performanță cât mai ridicată. Deseori, *limbajele* și *ustensilele* de înalt nivel vor permite o implementare și dezvoltare mai rapidă, cu o curbă de învățare relativ redusă. Există și reversul medaliei, și anume "*dacă vrei să faci ceva cum trebuie, să faci singur*". În acest sens, există limbaje care permit, nu fără un oarecare efort, implementarea de sisteme de **RV**, fără a fi neaparat destinate acestui scop, și care

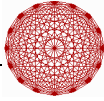


lasă dezvoltatorul să efectueze optimizările pe care le crede de cuviință. *Biblioteca grafică OpenGL* (GL de la **Graphic Library**; <http://www.opengl.org/>) oferă o interfață simplă și directă pentru operațiile fundamentale de randare 3D. Pe un nivel superior de abstractizare, se situează **OpenIV** (IV de la **Inventor**) și **Java3D** al caror model al programării permite efectuarea de sarcini dificile cum ar fi *traversarea grafului scenei, gestionarea modificărilor atributelor stărilor* etc., simplificând astfel sarcina aplicației. În particular, sunt disponibile implementări **Java3D** care utilizează **Direct3D** (Microsoft Corp. 2000) și **OpenGL**, **OpenIV** fiind de la bun început bazat pe **OpenGL**. De asemenea, **VRML** (Virtual Reality Modeling Language; <http://www.vrml.org>) – *Limbaaj de Modelare a Realității Virtuale*, din punct de vedere tehnic, nu este nici *realitate virtuală* și nici *limbaaj de modelare*. Realitatea virtuală implică o *experiență imersivă 3D*, cum ar fi cea obținută cu ajutorul *display-urilor* instalate pe capul utilizatorilor, și un *dispozitiv de interacțiune 3D*, cum este cazul binecunoscutelor, *mănuși digitale*. **VRML** nici nu necesită și nici nu restricționează *imersiunea*. Un *limbaaj de modelare* adevărat ar trebui să conțină mult mai multe *primitive* și *mecanisme de modelare geometrică*. **VRML** oferă, din acest punct de vedere, un minimum necesar de *facilități de modelare geometrică* având o serie de facilități care depășesc însă scopul unui limbaaj de modelare (*senzori, gestiunea evenimentelor*, etc). **VRML** la originea sa este un format simplu de fișiere, utilizat în schimbul de informații 3D între aplicații specifice, un analog 3D al *HTML-ului*, pe de alta parte, **VRML** oferă tehnologia care integrează cele 3 dimensiuni, două dimensiuni, *textul* și *multimedia* într-un model coerent, fundamental spațiului virtual.

3. Biblioteca grafică OpenGL

Biblioteca grafică OpenGL elaborată în limbajul **C++** este una dintre cele mai utilizate biblioteci ce oferă un număr mare de *funcții/primitive grafice* necesare în crearea *aplicațiilor grafice interactive* și care asigură o interfață independentă de platforma hardware. Scrisă în anul 1992, biblioteca grafică **OpenGL** implementează un *standard de grafică multiplatformă* având *stabilitate, portabilitate și fiabilitate*, cu o documentație tehnică adecvată, cu exemple de cod și informații utile accesibile free pe Internet (www.opengl.org). **OpenGL** este intens folosită pentru o mare varietate de platforme hardware și pentru o mare varietate de aplicații grafice 2D și 3D, folosite în cele mai variate domenii, de la proiectarea asistată de calculator (CAD), la modelare-simulare și realitate virtuală.

Aplicațiile **OpenGL** se execută pe platforme foarte variate și sub cele mai cunoscute sisteme de operare: **Windows 98/2000, Windows NT, Mac OS, Unix, Linux**. De asemenea, funcțiile **OpenGL** se pot apela din limbajele de programare **Ada, C, C++ și Java**. **OpenGL** este o *interfață software* ce include câteva sute de proceduri și funcții care permit programatorului să specifice obiecte și operațiile asupra lor implicate în producerea imaginilor grafice de înaltă calitate, în special imagini ale corpurilor 3D. **OpenGL** integrează grafica 3D în sistemul de ferestre (i.e. X sau Windows/NT). Biblioteca **OpenGL** desenează/vizualizează *primitive geometrice* (puncte, linii, poligoane) în diferite moduri selectabile. Primitivele geometrice sunt definite printr-un grup de unul sau mai multe vârfuri (*vertices*). Un *vârf* definește un



punct, extremitatea unei linii, vârful unui poligon. Fiecare vârf are asociat un set de date: *coordonate, culoare, normala, coordonate de textura*. **OpenGL** afișează *primitivele geometrice* într-un cadru *buffer*, acesta putând funcționa într-o serie de moduri selectabile. Fiecare *primitivă geometrică* este un punct, un segment de dreaptă, un poligon, un dreptunghi de pixeli sau chiar un *bitmap*. Fiecare mod poate fi schimbat în mod independent, setările unuia neafectând setările altuia, chiar dacă o serie de moduri interacționează în vederea detectării eventualelor depășiri din cadrul *buffer*. Modurile sunt setate, primitivele sunt specificate și alte operații ale **OpenGL** sunt descrise prin transmiterea de comenzi sub forma apelurilor de *funcții* sau *proceduri*.

Primitivele geometrice (punctele, segmentele de dreaptă și poligoanele) sunt definite prin intermediul unui grup de unul sau mai multe *vârfuri*. Un *vârf* definește un punct, un punct terminal al unei muchii sau un punct comun a două muchii ale unui poligon. Modelul de interpretare al comenzilor **OpenGL** este *client-server*. Efectele comenzilor **OpenGL** asupra cadrului *buffer* sunt, în cele din urmă, controlate de sistemul de ferestre, care realizează alocarea resurselor acestuia. Acest sistem determină porțiunile din cadrul *buffer* care pot fi accesate de **OpenGL** în orice moment și care transmite **OpenGL**-ului modul de structurare al acestor porțiuni.

Figura 5 prezintă o diagramă schematică a **OpenGL**. Majoritatea comenzilor pot fi cumulate într-o listă de afișare, *display list*, pentru o procesare ulterioară. Comenzile sunt trimise efectiv prin *pipeline-ul OpenGL*. Prima etapă oferă o modalitate eficientă de aproximare a *geometriei curbilor și suprafețelor*, prin evaluarea *funcțiilor polinomiale* în valorile prezente la intrare. Următoarea etapă operează asupra *primitivele geometrice* descrise prin *puncte, segmente de dreaptă și poligoane*. În cadrul acestei etape, *vârful* sunt transformate și iluminate, iar primitivele sunt decupate în raport cu volumul de vedere, ca pregătire a etapei de *rasterizare*. *Rasterizarea* produce o serie de *adrese și valori* în cadrul *buffer*, utilizând o descriere bidimensională a unui punct, segment de dreaptă sau poligon. Fiecare *fragment* produs astfel este trimis următoarei etape, care efectuează operații asupra fragmentelor individuale, înainte ca acestea să afecteze definitiv cadrul *buffer*.

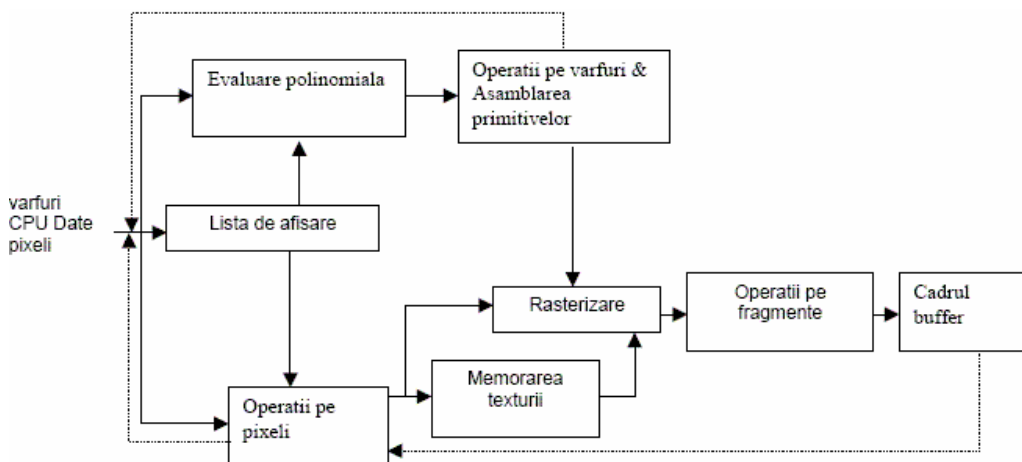
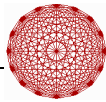


Figura 5. Diagrama de prelucrare OpenGL

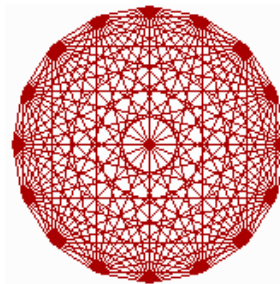


Funcțiile OpenGL execută secvențe de operații grafice asupra fiecărei primitive geometrice, definită printr-o mulțime de vârfuri și tipul acesteia. Coordonatele unui vârf (în 2D, 3D sau în coordonate omogene) sunt transmise prin apelul unei funcții `glVertex#()`. O primitivă geometrică se definește printr-o mulțime de vârfuri (descrierea geometrică) și printr-unul din tipurile prestabilite ce indică topologia de conectare a punctelor. Funcțiile `glBegin()` și `glEnd()` delimitează o mulțime de comenzi. Tipurile de primitivă geometrică sunt: `GL_POINTS`, `GL_LINES`, `GL_LINE_STRIP`, `GL_LINE_LOOP`, `GL_TRIANGLES`, `GL_TRIANGLE_STRIP`, `GL_TRIANGLE_FAN`, `GL_QUADS`, `GL_QUADS_STRIP`, `GL_POLYGON`. Generatoarele de imagini mai reprezentative sunt: *ProVision* (compania Division), *Evens&Sutherland*, *Thomson CSF*, stațiile grafice *Silicon Graphics*. Exemplificăm mai jos, prin două exemple, utilizarea funcțiilor și primitivelor geometrice **OpenGL**.

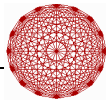
```
// glex.cpp
#include <windows.h>
#include <gl\gl.h>
#include <gl\glu.h>
#include <gl\glaux.h>
#include <math.h>
#define PI 3.141592
int n=16; // numar de varfuri
double radius = 20; // raza cercului
glColor3d(0,0,0); // culoare
glBegin (GL_LINE_LOOP); // apel primitiva geometrica
  for (int i=0; i<n; i++ )
  { double angle = 2 * PI*I / n; // unghiul corespunzator laturei
    glVertex3d (radius * cos (angle) , radius * sin (angle), -15);
  }
glEnd();
```

Exemplu. Generarea
unui poligon regulat
în spațiu 3D

```
// glsigla_CNIV.cpp ; desenarea siglei CNIV-background
#include <windows.h>
#include <gl\gl.h>
#include <gl\glu.h>
#include <gl\glaux.h>
#include <math.h>
#define PI 3.141592
int n=16; // numar de varfuri
double radius = 50; // raza cercului
glColor3d(0,0,0); // culoare
glBegin (GL_LINES); // apel primitiva
geometrica
  for (int i=0; i<n; i++ )
  { double angle_1 = 2 * PI* I / n;
    for (int j=i+1; j<n+1; i++);
    {double angle_2 = 2 * PI * J / n;
      glVertex2d (radius * cos(angle_1), radius * sin(angle_1));
      glVertex2d (radius * cos(angle_2), radius * sin(angle_2));
    }
  }
glEnd();
```

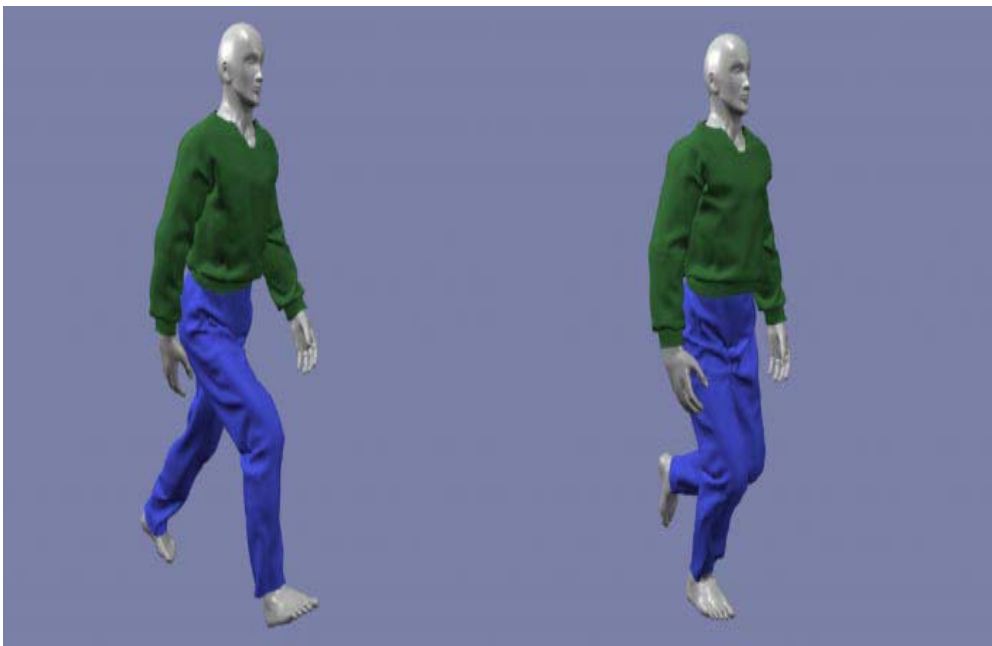


Generarea imaginilor vizuale în realitatea virtuală cuprinde:



- *crearea modelului scenei virtuale* – proces *off-line* de creare a colecției de modele ale obiectelor tridimensionale ce va reprezenta mediul virtual; *toolkit*-uri pentru generarea obiectelor și scenelor virtuale sunt: **3D Studio**, **Autocad**, **AC3D**, **Sense8**, **Designer Workbench** etc.
- *vizualizarea scenei virtuale* – proces *on-line* desfășurat în timp real în care scena virtuală este explorată interactiv, imaginea scenei fiind vizualizată pe display și depinde de condițiile de explorare (poziție de observare, acțiuni interactive etc.); *toolkit*-uri pentru redarea imaginii scenelor virtuale sunt: **Performer**, **EasyScene**, **Browser CosmoPlayer** etc.

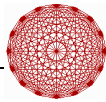
Generatoarele de imagini mai reprezentative sunt: *ProVision* (compania Division), *Evens&Sutherland*, *Thomson CSF*, stațiile grafice *Silicon Graphics*.



(David Baraff, Andrew Witkin, *Large Steps in Cloth Simulation*, SIGGRAPH 98, COMPUTER GRAPHICS Proceedings, Annual Conference Series, 1998)

4. VRML și WWW

Limbajul **VRML** (*Virtual Reality Modeling Language*) este limbaj pentru crearea scenelor virtuale tridimensionale cu posibilitatea de acces a acestora prin rețeaua *Internet* folosind protocolul **WWW**. Compania **Silicon Graphics Inc.** a inițiat în anul 1989 un proiect pentru proiectarea și construirea unui *mediu de dezvoltare* necesar *aplicațiilor grafice interactive 3D* și de a utiliza acest mediu pentru a construi un nou suport al interfeței **3D**. Prima fază a proiectului s-a concentrat asupra proiectării și construirii semanticii și mecanismului fundamentării cadrului general. În anul 1992 s-a lansat *pachetul de ustensile* **Iris Inventor 3D** ca prim rezultat al acestor eforturi. **Iris**



Inventor era un pachet **C++** care definea o bună parte din semantica regăsită în **VRML**-ul de azi (www.vrml.org). O importantă parte a pachetului o constituia *formatul fișierului* utilizat în stocarea obiectelor aplicației. Versiunea 2.0 este numită **VRML97** fiind standard ISO din anul 1997 cu facilități importante privind utilizarea **WWW**. În anul 2000 apare o extindere numită **GeoVRML** privind reprezentarea datelor geografice de mari dimensiuni. Un *fișier VRML* conține descrierea unei *scene virtuale* și a unor acțiuni interactive, fiind accesat și de un program navigator (*browser*).

Fișierele **VRML** descriu obiecte și lumi 3D utilizând un *graf ierarhic* al scenei. Entitățile din grafurile scenei se numesc *noduri*, **VRML** 2.0 definind 54 de tipuri diferite de noduri, incluzând *primitive geometrice*, *proprietăți de aspect*, *sunet* și *proprietăți ale sunetului*, și o serie de tipuri de noduri de grupare. Grafurile scenei **VRML**, asemeni grafurilor scenei **OpenIV**, este un graf aciclic orientat. Nodurile pot conține alte noduri, unele tipuri de noduri putând avea descendenți (*children*), și pot fi conținute în unul sau mai multe noduri, având astfel mai mult de un părinte (*parent*). Cu toate acestea, un nod nu se poate include pe el însuși.

```
Group{
  children[
    Shape{
      geometry Box{. . .}
      appearance Appearance{. . .}
    }
    Transform{. . .}
    . . .
  ]
}
```

Un fișier **VRML** este format conceptual din 4 elemente definitorii: *header*, *prototip*, *graf al scenei* și *mecanism de rutare a evenimentelor*. Aceste elemente sunt analizate și executate de către *browserul VRML* în momentul reprezentării scenei. *Header-ul* este prima linie a fișierului care reprezintă identificatorul formatului **VRML**:

```
#VRML V1.0 ascii pentru versiunea 1.0
#VRML V2.0 utf8 pentru versiunea 2.0
```

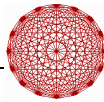
Șirurile **ascii** și **utf8** specifică tipul de codare al datelor din fișierului **VRML**. *Header-ul* trebuie să conțină exact un singur spațiu între șiruri pentru ca el să fie recunoscut de către browser. În structura unui fișier **VRML**, după linia antet, putem întâlni combinații ale următoarelor instrucțiuni: declarații **PROTO** sau **EXTERNPROTO**; declarații de noduri; instrucțiuni **USE**; instrucțiuni **ROUTE**.

Formatul declarației *nodurilor* în **VRML** are forma generală :

```
[DEF <name>] <nodeType> { <body> }
```

Singurul parametru care nu trebuie să lipsească dintr-o declarație este **nodType** care poate fi unul din nodurile implementate de **VRML**. Limbajul oferă posibilitatea redenumirii nodurilor standard cu ajutorul cuvântului cheie **DEF** și referirea lor ulterioară cu ajutorul lui **USE**.

```
DEF FrameRod Shape {
  appearance Appearance {
    material Material {
      diffuseColor .2 .06 .02
    }
  }
  geometry Cylinder { radius 0.05 }
```



```
}  
#...  
Transform {  
    translation 0 0 0  
    children USE FrameRod  
}
```

În corpul nodului pot fi incluse declarații de câmpuri, instrucțiuni **IS**, **ROUTE**, **PROTO** sau **EXTERNPROTO** într-o ordine aleatoare.

Fiecare nod are următoarele caracteristici:

- *un nume de tip* - de exemplu, **Box**, **Color**, **Group**, **Sphere**, **Sound**, **Spotlight**;
- *câmpuri care arată modul în care fiecare nod diferă de alt nod de același tip*; valorile câmpurilor sunt păstrate în fișiere **VRML** alături de noduri și codifică starea lumii virtuale;
- *un set de evenimente ce pot fi receptate și transmise* - fiecare nod poate primi mai multe tipuri de *evenimente* ce vor produce modificări în starea nodului și de asemenea, poate să genereze evenimente ca răspuns la modificarea stării nodului;
- *o implementare* - implementarea fiecărui nod definește modul în care reacționează la evenimentele pe care le poate primi, când generează evenimentele și aparițiile vizuale sau auditive ale acestora; declarația **PROTO** poate fi folosită la definirea unor noi tipuri de noduri, cu caracteristici definite în caracteristicile (comportamentele) altor noduri;
- *un nume* - nodurile pot fi numite; numele acestora poate fi folosit în alte declarații, la referirea unei instanțe specifice nodului.

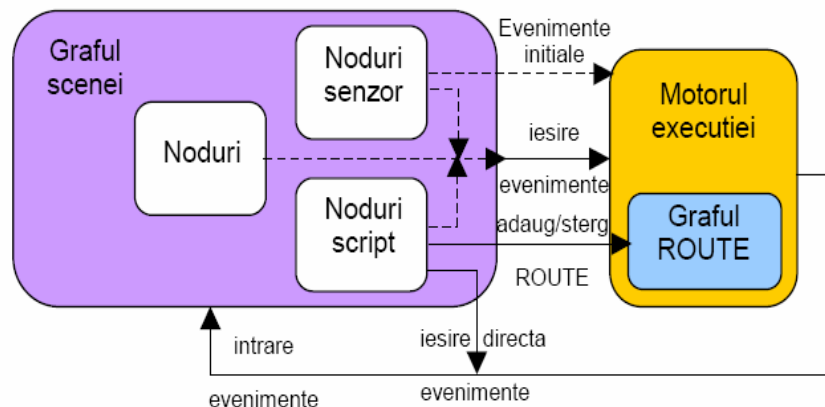
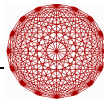


Figura 6. Modelul conceptual de execuție

Fișierele **VRML** pot conține referințe la fișiere existente în diferite alte formate standard. Astfel, fișierele **JPEG**, **PNG**, **GIF**, și **MPEG** pot fi utilizate ca hărți de textură aplicate obiectelor **VRML**. Fișierele **WAV** și **MIDI** pot fi utilizate în specificarea sunetelor emise în lume. Fișierele conținând cod **Java** sau **JavaScript** pot fi referite și utilizate pentru implementarea comportamentului programat al obiectelor din univers. Fiecare dintre acestea constituie un standard independent, ales a fi utilizat împreună cu **VRML** datorită largii sale utilizări în Internet. Există cel puțin șase modalități diferite de a combina **VRML**, **HTML** și **Java**, fiind astfel destul de greu de identificat modul

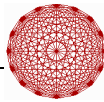


de combinare cerut de standardele **HTML**, **Java**, și/sau **VRML**. Următoarea listă conține o enumerare succintă a tuturor acestor posibilități:

- *fișier VRML interior unui fișier HTML* - aceasta este o parte a HTML utilizând tag-urile HTML `<EMBED>` sau `<OBJECT>`, cu toate că HTML nu necesită ca un browser HTML să suporte includerea fișierelor **VRML** (sau a oricarui alt tip de fișier) în documente HTML.
- *cod Java interior unui fișier VRML* - această soluție este standardizată ca parte a **VRML 2.0**, utilizând nodul **Script**, care se refera la cod compilat **Java**.
- *applet Java ce comunică cu un browser VRML* - aceasta este o extensie a **VRML 2.0** cunoscută ca *External Authoring Interface (EAI)*.
- *clase Java corespunzând nodurilor VRML* - o serie de companii au dezvoltat pachete de programe care definesc reprezentările nodurilor **VRML** în memorie și care pot fi utilizate în orice mod de către programator; acestea pot fi extrem de utile atunci când se implementează browsere sau ustensile **VRML**, fără însă a fi părți standard, nici pentru **VRML** și nici pentru **Java**.
- *fișier HTML interior unui fișier VRML* - utilizarea unui fișier HTML ca hartă de textură pentru a fi afișată într-o lume 3D, poate constitui o extensie interesantă a limbajului **VRML**.
- *applet Java interior unui fișier VRML* - utilizarea unui *applet Java* ca hartă a texturii pentru a afișa programul **Java** într-o lume 3D, ar putea constitui o extensie interesantă. Interacțiunea cu programul Java poate fi suportată prin proiectarea deplasării dispozitivului de indicare în *applet*.

Bibliografie

- [Alex00] M.Alexa, J.Behr, W.Müller: The Morph Node, pg. 29-34, VRML 2000, Monterey, USA, ACM 2000 1-58113-211-5.
- [Beer97] C.Beeson: An Object-Oriented Approach To VRML Development, pg. 17-24, VRML 97, Monterey USA, Copyright 1997 ACM 0-89791-886-x.
- [Bida99] R.Bidarra, W.F.Bronsvort: Validity Maintenance of Semantic Feature Models, pg.85-96, Fifth Symposium on Solid Modeling, Ann Arbor MI, ACM 1999 1-58113-080-5.
- [Bier00] A.D.Bierbaum: VR Juggler: A virtual platform for virtual reality application development, MS.Thesis, Iowa State University, 2000.
- [Brut98] D.Brutzman: The Virtual Reality Modeling Language and Java, Communications of the ACM, vol. 41 no. 6, June 1998, pp. 57-64.
- [Chen99] H.Chen, S.Fang: A Volumetric Approach to Interactive CSG Modeling and Rendering, pg. 318-319, Fifth Symposium on Solid Modeling, Ann Arbor MI, ACM 1999 1-58113-080-5.
- [Cruz95] C. Cruz-Neira. Virtual Reality Based on Multiple Projection Screens: The CAVE and Its Applications to Computational Science and Engineering. Ph.D. Dissertation, University of Illinois at Chicago. May 1995
- [Dixi97] H.Dixit, S.Kannapan, D.L.Taylor: 3D Geometric Simulation of MEMS Fabrication Processes: A Semantic Approach, pg. 376-387, Solid Modeling'97, Atlanta, USA, ACM 1997 0-89791-946-7.
- [Dorn00] Ralf Dörner, Paul Grimm: Three-dimensional Beans – Creating Web Content Using 3D Components In A 3D Authoring Environment, pg. 69-74, VRML 2000, Monterey, USA, ACM 2000 1-58113-211-5.
- [Edal99] A.Edalat, A.Lieutier: Foundation of a Computable Solid Modeling, pg. 278-284, Fifth Symposium on Solid Modeling, Ann Arbor MI, ACM 1999 1-58113-080-5.



- [Fish00] P.A.Fishwick: 3D Behavioral Model Design for Simulation and Software Engineering, pg. 7-16, VRML 2000, Monterey, USA, ACM 2000 1-58113-211-5.
- [Hall94] S. Halliday and M. Green, "A Geometric Modeling and Animation System for Virtual Reality." *Virtual Reality Software and Technology (VRST 94)*, 71-84, Singapore, August 1994.
- [Heis93] J.Heisserman, R.Woodbury: Generating Languages of Solid Models, 2nd ACM Solid Modeling 1993, Montreal, Canada, pg.103-113, 1993 ACM 0-89791-584-4.
- [Ione00] F.Ionescu: Grafica in realitatea virtuala, Ed.Tehnica, 2000, ISBN 973-31-1466-9
- [Java] Java3D API Specification, www.sun.com.
- [Jung99] B.Jung, J.T.Mildet: An Open Virtual Environment for Autonomous Agents Using VRML and Java, pg. 7-11, VRML99 Paderborn, Germany, ACM 1999 1-58113-079-1199.
- [Kuma97] V.Kumar, D.Dutta: An Approach to Modeling Multi-Material Objects, pg. 336-345, Solid Modeling'97, Atlanta, USA, ACM 1997 0-89791-946-7.
- [Mrob] MRObjects home page: <http://www.cs.ualberta.ca/~graphics/mrobjects/>
- [Mrto] MR Toolkit home page: <http://www.cs.ualberta.ca/~graphics/MRToolkit.html>
- [Parr95] T.J. Parr, T.F.Rohalyt: A Language for Creating and Manipulating VRML, VRML'95, San Diego USA, pg.123-131, 1995 ACM 0-89791-818-5. 164
- [Pete96] M.T.Peterson: 3D Studio MAX Fundamentals, New Riders Pub., 1996.
- [Popo00a] M. D. Popovici, M. I. Popovici: C++. Tehnologia orientată spre obiecte. Aplicații, Ed. Teora, București, 2000.
- [Popo02b] M.D. Popovici: Metode si tehnici pentru realizarea mediilor virtuale, Referat de doctorat, Universitatea "Politehnica" București, 2002
- [Popo02c] M.D. Popovici: Limbaje de modelare a obiectelor in mediilor virtuale, Referat de doctorat, Universitatea "Politehnica" București, 2002
- [Popo02d] M.D. Popovici: Realitatea virtuala: Tehnologii componente, Referat de doctorat, Universitatea "Politehnica" București, 2002
- [Rapo95] A.Rappoport: Geometric Modeling: a New Fundamental Framework and its Practical Implications, pg. 31-41, Solid Modeling' 95, Salt Lake City, Utah USA, 1995 ACM
- [Redd00] M.Reddy, L.Iverson, Y.G.Leclerc: Under the Hood of GeoVRML 1.0, pg. 23-29, VRML 2000, Monterey, USA, ACM 2000 1-58113-211-5.
- [Salm00] M.Salmela, H.Kyllönen: Smart Virtual Prototypes: Distributed 3D Product Simulations for Web Based Environments, VRML 2000, Monterey, USA, ACM 2000.
- [Scho00] B.Schonhage, A.van Ballegooij, A.Eliens: 3D Gadgets for Business Process Visualization - a case study. VRML 2000, Monterey USA, pg. 131-138, ACM 2000.
- [Sens] "WorldToolKit Release 8: Technical Overview", <http://www.sense8.com>
- [SGI01] Iris Performer Homepage. <http://www.sgi.com/Technology/Performer>
- [SGI02] Iris Performer Getting Started Guide, IRIS Insight. SGI online manual
- [Sig98] SIGGRAPH 98, COMPUTER GRAPHICS Proceedings, Annual Conference Series, A Publication of ACM SIGGRAPH, 1998
- [Shap95] V.Shapiro, D.L.Vossler: What is a parametric family of solids?, pg. 43-54, Solid Modeling '95, Salt Lake City, Utah USA, 1995 ACM 0-89791-672 -7.
- [Shaw93a] C. Shaw, M. Green, J. Liang, and Y. Sun, "Decoupled Simulation in Virtual Reality with the MR Toolkit." ACM Transactions on Information Systems, Vol. 11, Number 3,1993.
- [Shaw93b] C. Shaw and M. Green, "The MR Toolkit Peers Package and Experiment." *IEEE Virtual Reality Annual International Symposium (VRAIS ' 93)*, 463-469, September 1993.
- [Vla92] M. Vlada ș.a., Grafică pe calculator în limbajele Pascal și C, Editura Tehnică, vol. I Implementare, vol. II Aplicații, ISBN 973-31-0406-X , 1992.
- [Vla04] M. Vlada, Birotică: Tehnologii multimedia, Editura Universității din București, ISBN 973-575-847-4, 2004
- [VRML] VRML97 Standard, www.vrml.org, www.web3D.org
- [Wern94] J.Werneck: The Inventor Mentor, Programming Object-Oriented 3D Graphics with Open Inventor, Addison-Wesley, 1994, ISBN 0-201-62495-8.